A Current-Mode DRAM for CVNS

Report of A Work In Progress

Golnar Khodabandehloo

Dr. M. Ahmadi

Dr. M. Mir

June 13, 2008

Outline

- Motivation
- CVNS
  - Introduction
  - Digit Generation
  - Error Correction
- Current-Mode DRAM
  - Introduction
  - Storage Cell
- DRAM for CVNS
  - Circuits, Simulations results
    - XOR
    - D-Latch
    - CVNS to Binary
    - Binary to CVNS
    - Refresh Circuit
- Future Work

Motivation

- Memory Design
  - Compactness
  - More than 1 bit storage on each cell
  - Low refreshing rate
  - Fast read/write

- CVNS, a potential candidate

Introduction to CVNS

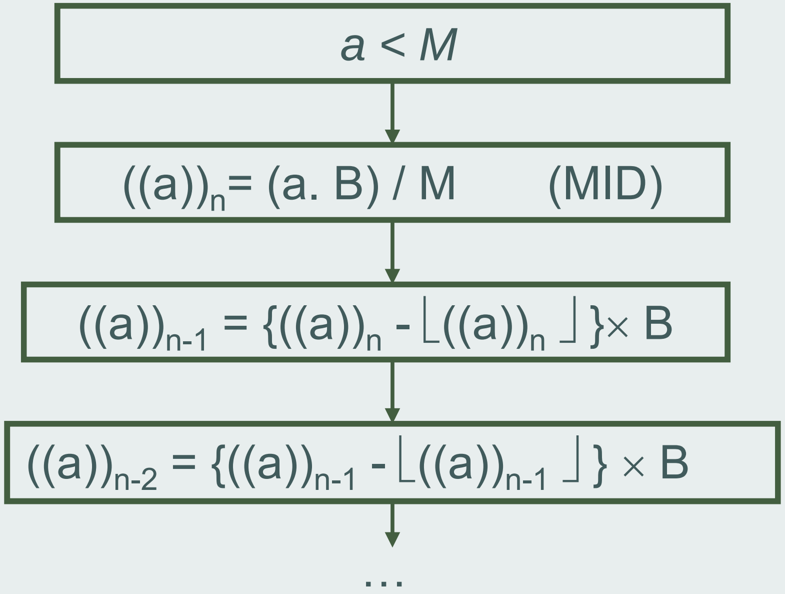
- Continuous Valued Number System
- Non-integer modular arithmetic with positive radix
- Uses classical analog circuits for implementation
- Analog arithmetic advantages
    - High Speed
    - Limited interconnections
    - Reduced Area
    - Reduced system and cross-talk noise
- Allows for arbitrary accuracy of arithmetic circuits

Digit Generation in CVNS

- Cascade Digit Generation

Calculates the radix- B CVNS digits in serial, starting from MID

- Modular Digit Generation



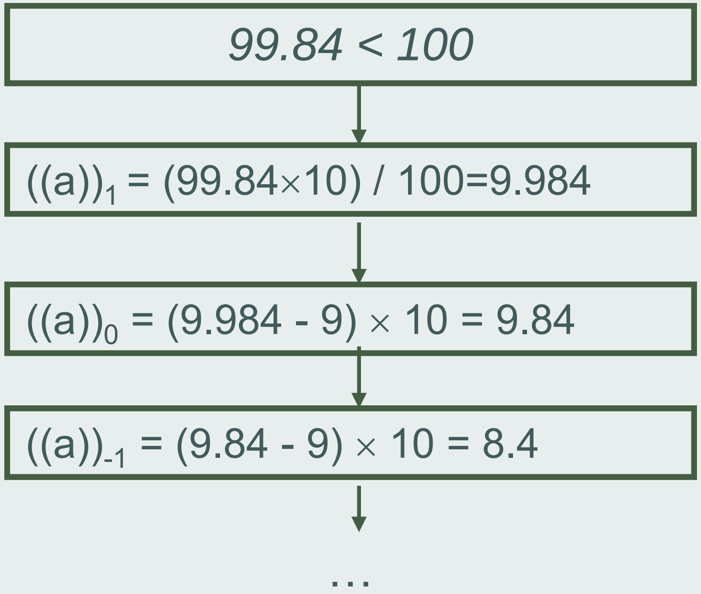
$a < M$

$((a))_n = (a.\ B) / M \quad \text{(MID)}$

$((a))_{n-1} = \{((a))_n - \lfloor ((a))_n \rfloor\} \times B$

$((a))_{n-2} = \{((a))_{n-1} - \lfloor ((a))_{n-1} \rfloor\} \times B$

…



$99.84 < 100$

$((a))_1 = (99.84 \times 10) / 100 = 9.984$

$((a))_0 = (9.984 - 9) \times 10 = 9.84$

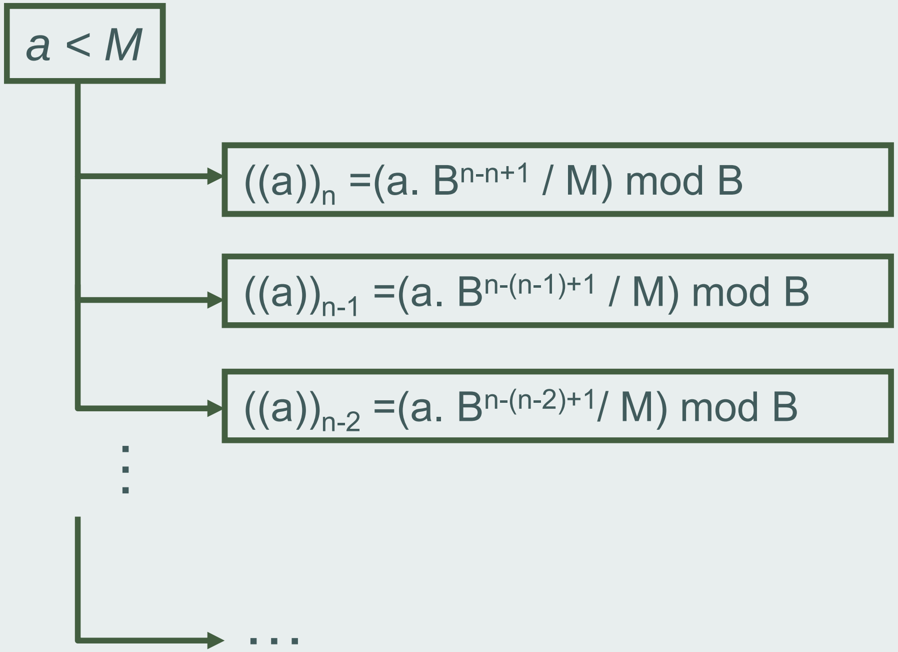
$((a))_{-1} = (9.84 - 9) \times 10 = 8.4$

…

Digit Generation in CVNS

- Cascade Digit Generation
- Modular Digit Generation

Computes all of the radix- B CVNS digits in parallel, independent of each other



$a < M$

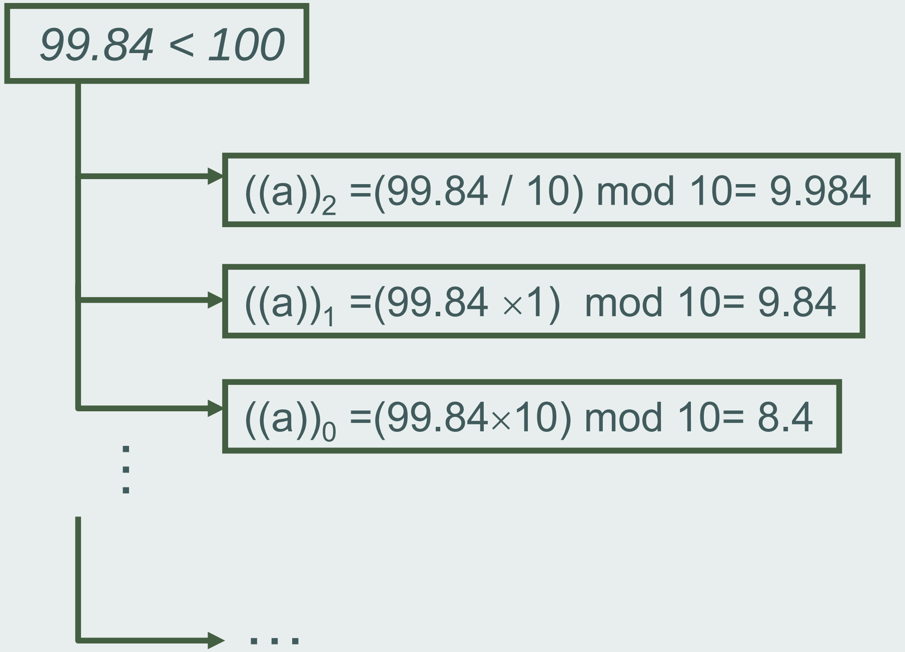
$((a))_n = (a.\ B^{n-n+1} / M) \bmod B$

$((a))_{n-1} = (a.\ B^{n-(n-1)+1} / M) \bmod B$

$((a))_{n-2} = (a.\ B^{n-(n-2)+1} / M) \bmod B$

⋮

…



$99.84 < 100$

$((a))_2 = (99.84 / 10) \bmod 10 = 9.984$

$((a))_1 = (99.84 \times 1) \bmod 10 = 9.84$

$((a))_0 = (99.84 \times 10) \bmod 10 = 8.4$

⋮

…

$(\alpha)\bmod \beta = \alpha - \beta \lfloor \alpha / \beta \rfloor$

Error Correction in CVNS

- Digit Recovering
    - Chain-like relationship between digits (overlap)

- Reverse Evolution (RE) Method
    - Increases the accuracy of analog digits
    - A sequential process

Error Correction in CVNS (Cont.)

- RE method

$((a))_i$ (original digit) $\quad k \le i \le n$

$((a'))_i$ (errored digit) $= ((a))_i + \varepsilon_i$

$$[((a'))_i\,]_R = [((a'))_i - ((a''))_{i-1} / B]_R \bmod^{+} B \qquad ((a')) \ge 0$$

$$[((a'))_i\,]_R = [((a'))_i - ((a''))_{i-1} / B]_R \bmod^{-} B \qquad ((a')) < 0$$

$$((a))''_i \text{ (corrected digit)} = \begin{cases} [((a'))_i]_R + ((a''))_{i-1} / B & i > K \\ ((a'))_i & i = K \end{cases}$$

- $(a) \bmod^{+} B = (a \bmod B + B) \bmod B \quad 0 \le (a) \bmod^{+} B < B$

- $(a) \bmod^{-} B = (a \bmod B - B) \bmod B \quad -B < (a) \bmod^{-} B \le B$

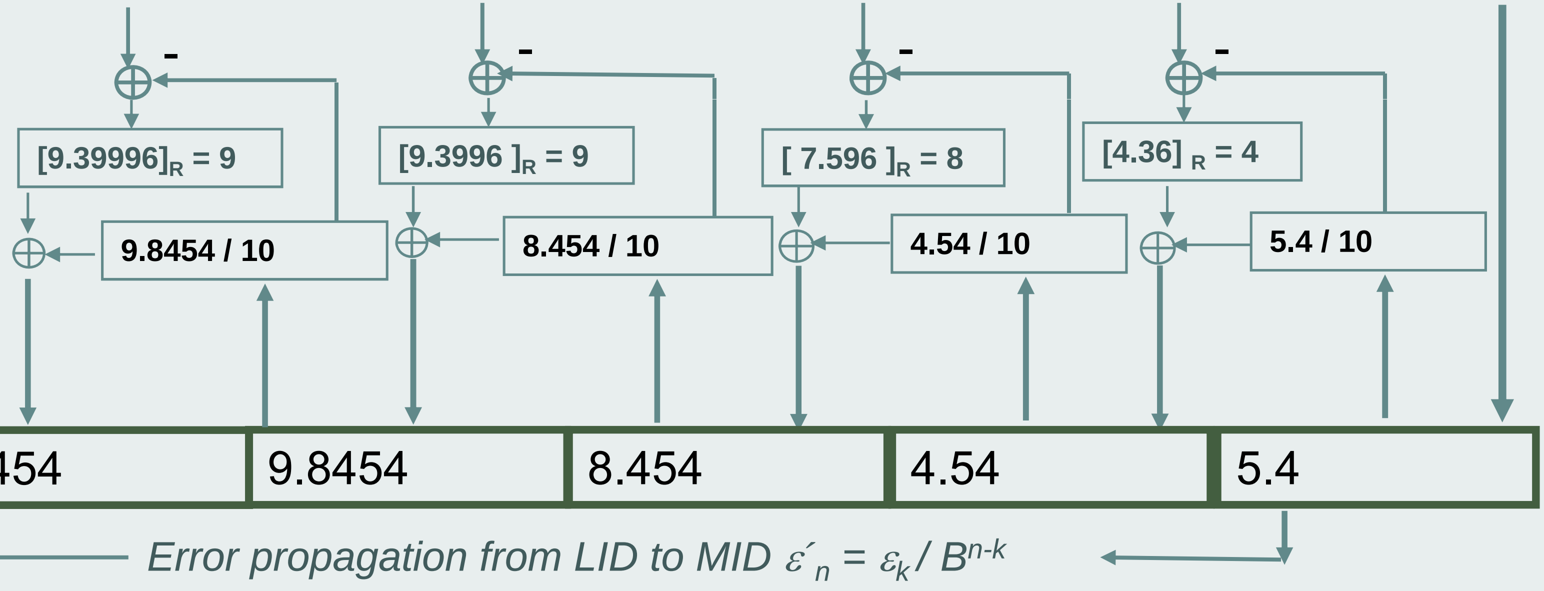
Error Correction in CVNS (Cont.)

- Example for a random value; a = 9.9845, B= 10, error=±0.4

i	2	1	0	-1	-2
$((a))_i$	9.9845	9.845	8.45	4.5	5

Error is introduced to each digit

$((a'))_i$	10.3845	10.245	8.05	4.9	5.4



9.98454	9.8454	8.454	4.54	5.4

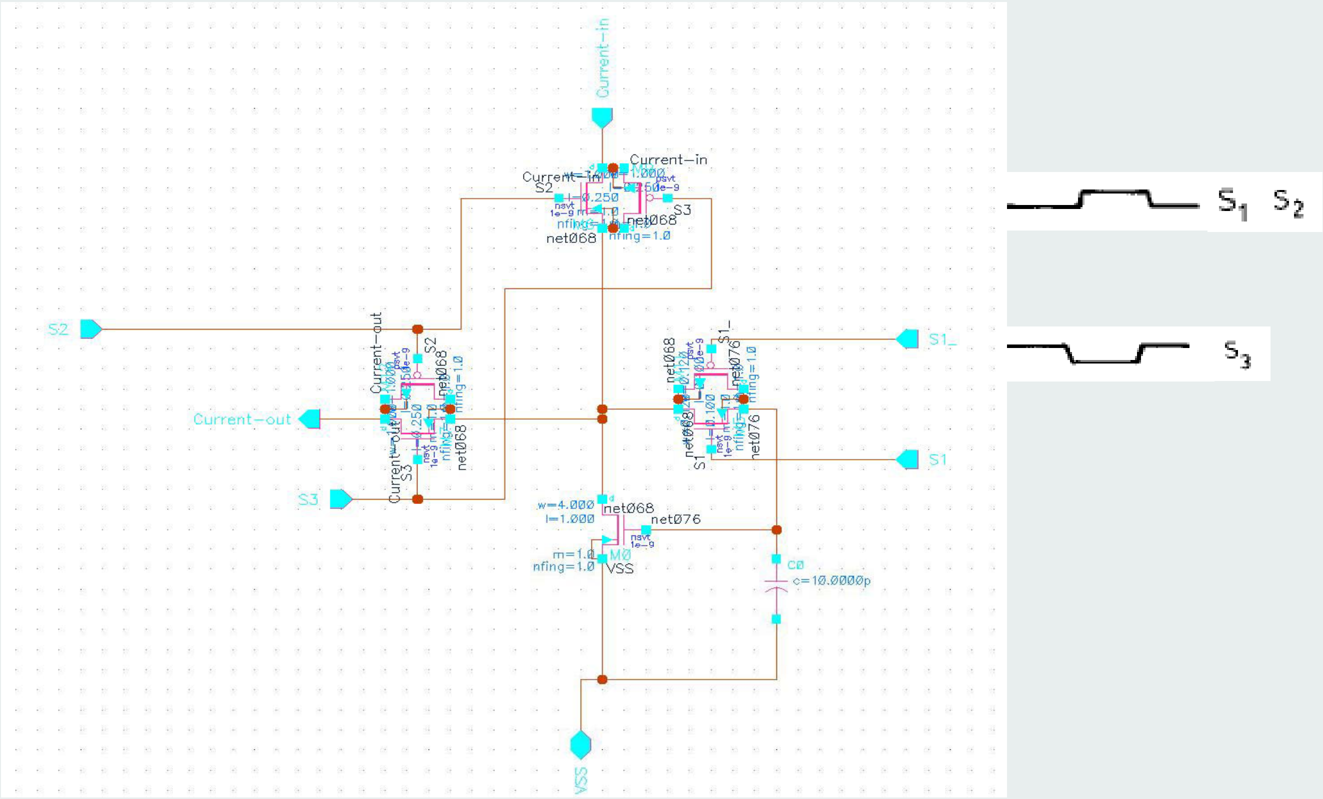
Error propagation from LID to MID $\varepsilon'_n = \varepsilon_k / B^{n-k}$

Introduction to Current-Mode DRAM

- Dynamic Random Access Memory
  - Memory cells requiring constant refreshing
    - Capacitor
    - Transistor
- Current-Mode DRAM
  - Fast sensing speed for the stored values
  - Capability of operating at lower power supply voltages

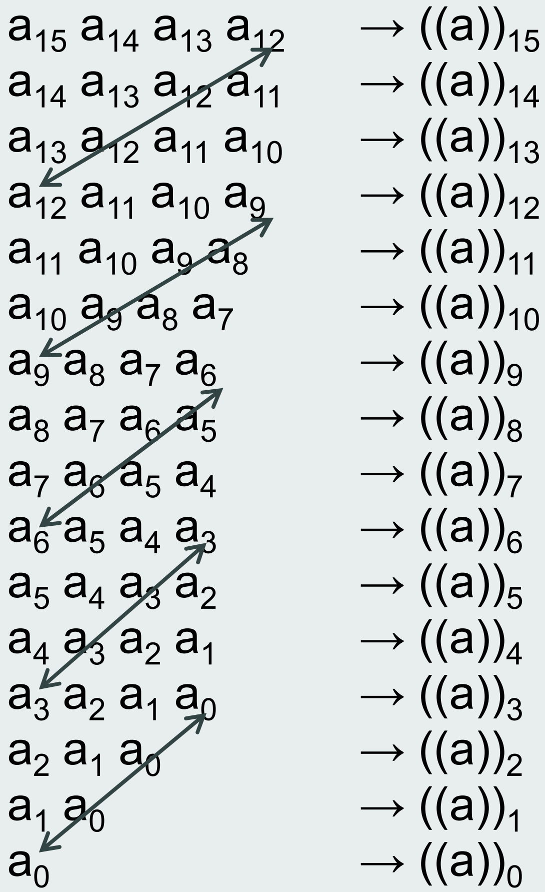
Storage Cell in Current-Mode DRAM

- Current Copier Cell as a Storage Element



Current-Mode DRAM for CVNS

- Correction Using CVNS



$a_{15}\ a_{14}\ a_{13}\ a_{12} \rightarrow ((a))_{15}$

$a_{14}\ a_{13}\ a_{12}\ a_{11} \rightarrow ((a))_{14}$

$a_{13}\ a_{12}\ a_{11}\ a_{10} \rightarrow ((a))_{13}$

$a_{12}\ a_{11}\ a_{10}\ a_{9} \rightarrow ((a))_{12}$

$a_{11}\ a_{10}\ a_{9}\ a_{8} \rightarrow ((a))_{11}$

$a_{10}\ a_{9}\ a_{8}\ a_{7} \rightarrow ((a))_{10}$

$a_{9}\ a_{8}\ a_{7}\ a_{6} \rightarrow ((a))_{9}$

$a_{8}\ a_{7}\ a_{6}\ a_{5} \rightarrow ((a))_{8}$

$a_{7}\ a_{6}\ a_{5}\ a_{4} \rightarrow ((a))_{7}$

$a_{6}\ a_{5}\ a_{4}\ a_{3} \rightarrow ((a))_{6}$

$a_{5}\ a_{4}\ a_{3}\ a_{2} \rightarrow ((a))_{5}$

$a_{4}\ a_{3}\ a_{2}\ a_{1} \rightarrow ((a))_{4}$

$a_{3}\ a_{2}\ a_{1}\ a_{0} \rightarrow ((a))_{3}$

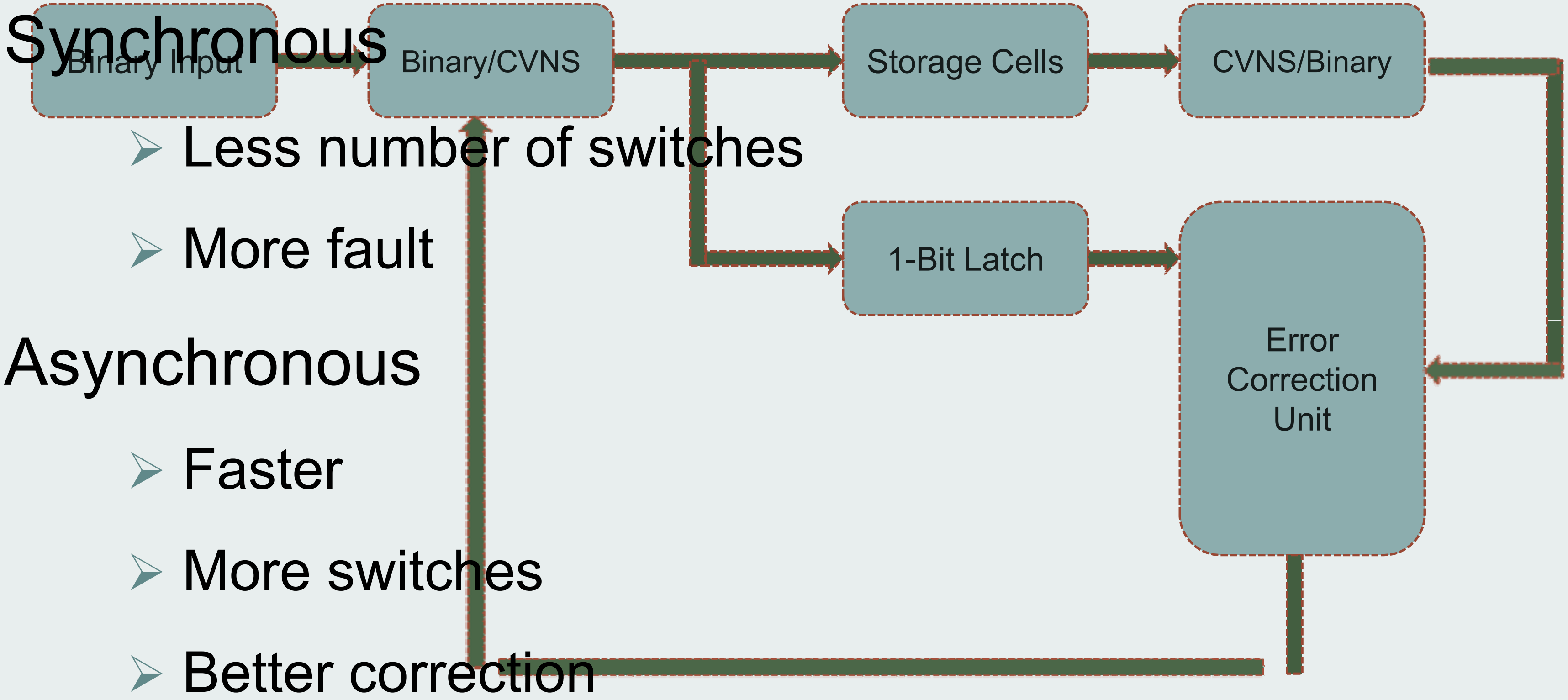
$a_{2}\ a_{1}\ a_{0} \rightarrow ((a))_{2}$

$a_{1}\ a_{0} \rightarrow ((a))_{1}$

$a_{0} \rightarrow ((a))_{0}$

Current-Mode DRAM for CVNS (Cont.)

- Block Diagram
- Synchronous
  - Less number of switches
  - More fault
- Asynchronous
  - Faster
  - More switches
  - Better correction



Binary Input

Binary/CVNS

Storage Cells

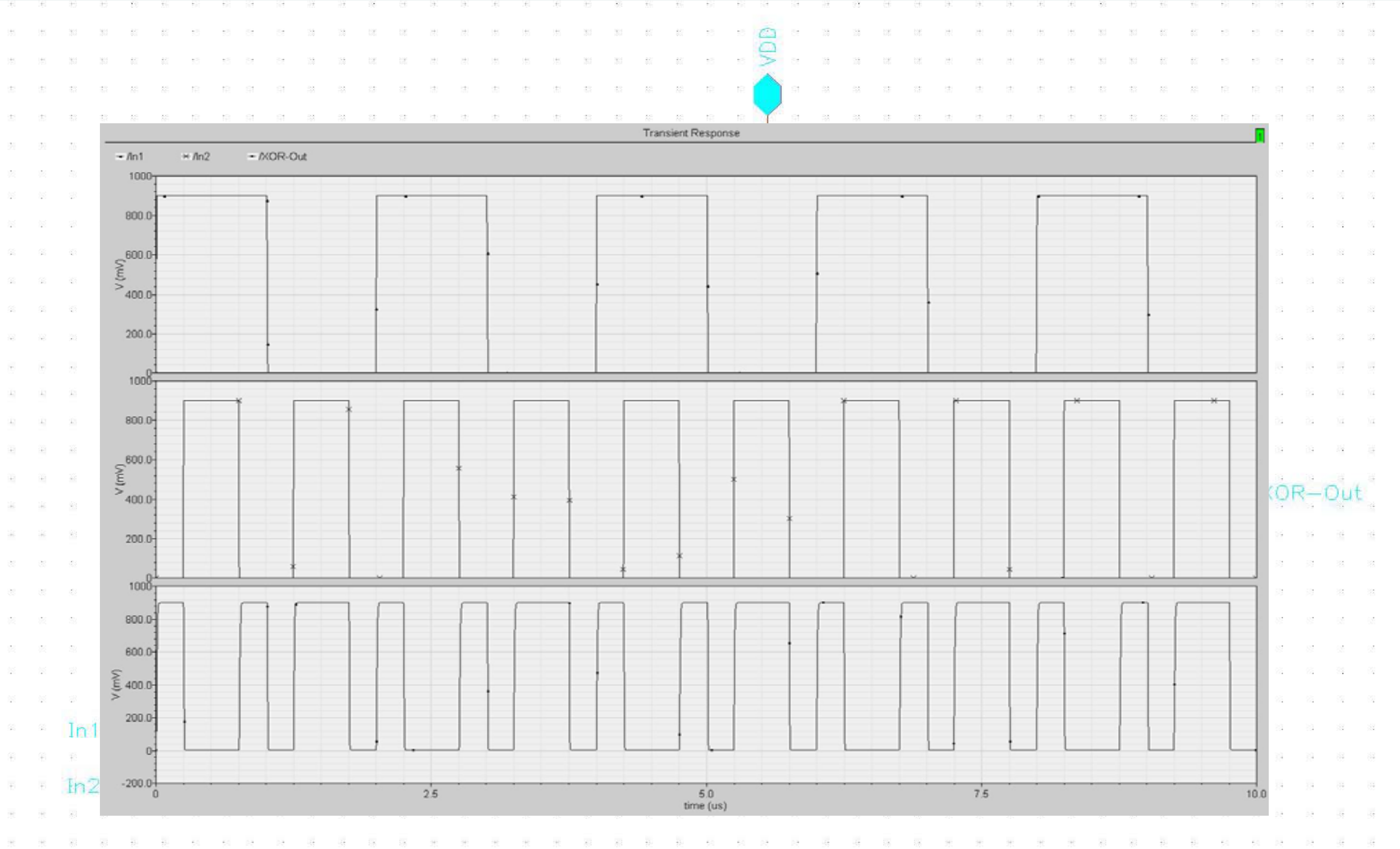
CVNS/Binary

1-Bit Latch

Error Correction Unit

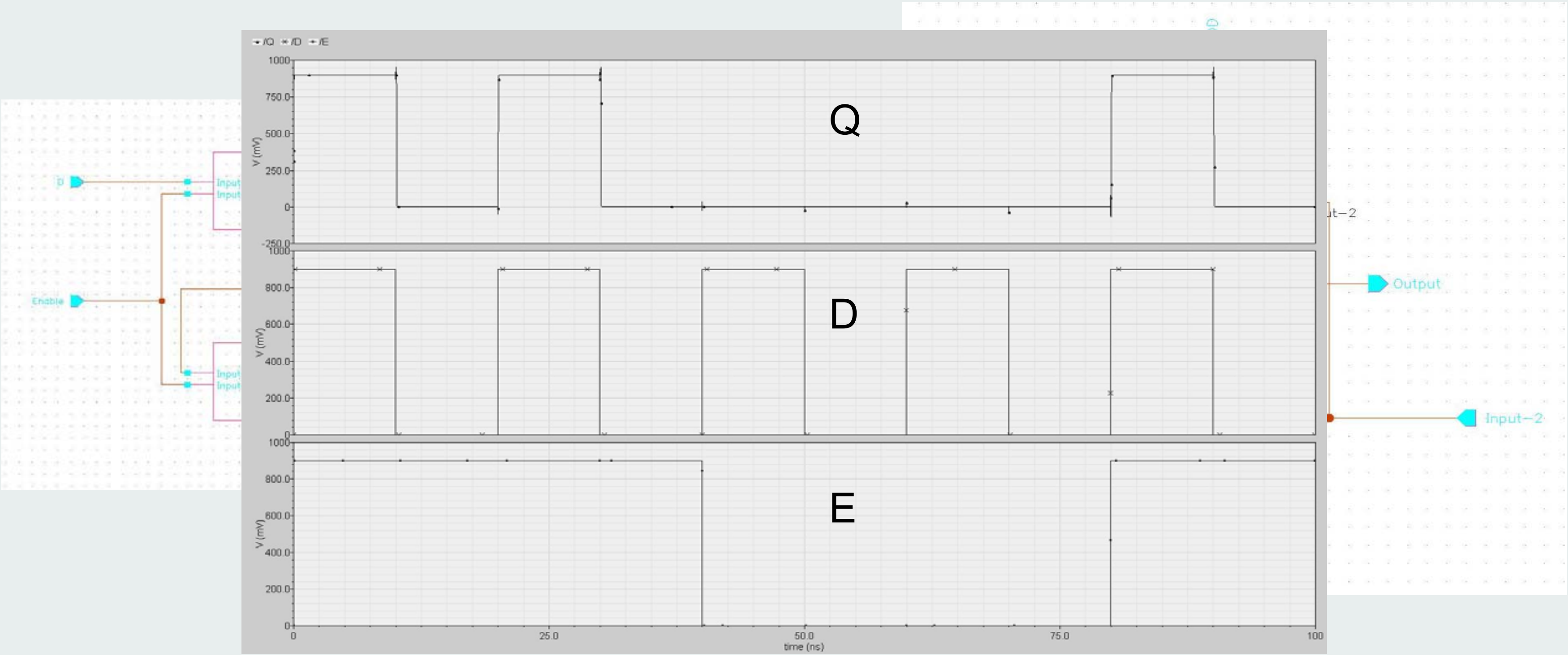
XOR

- Circuit
- Simula



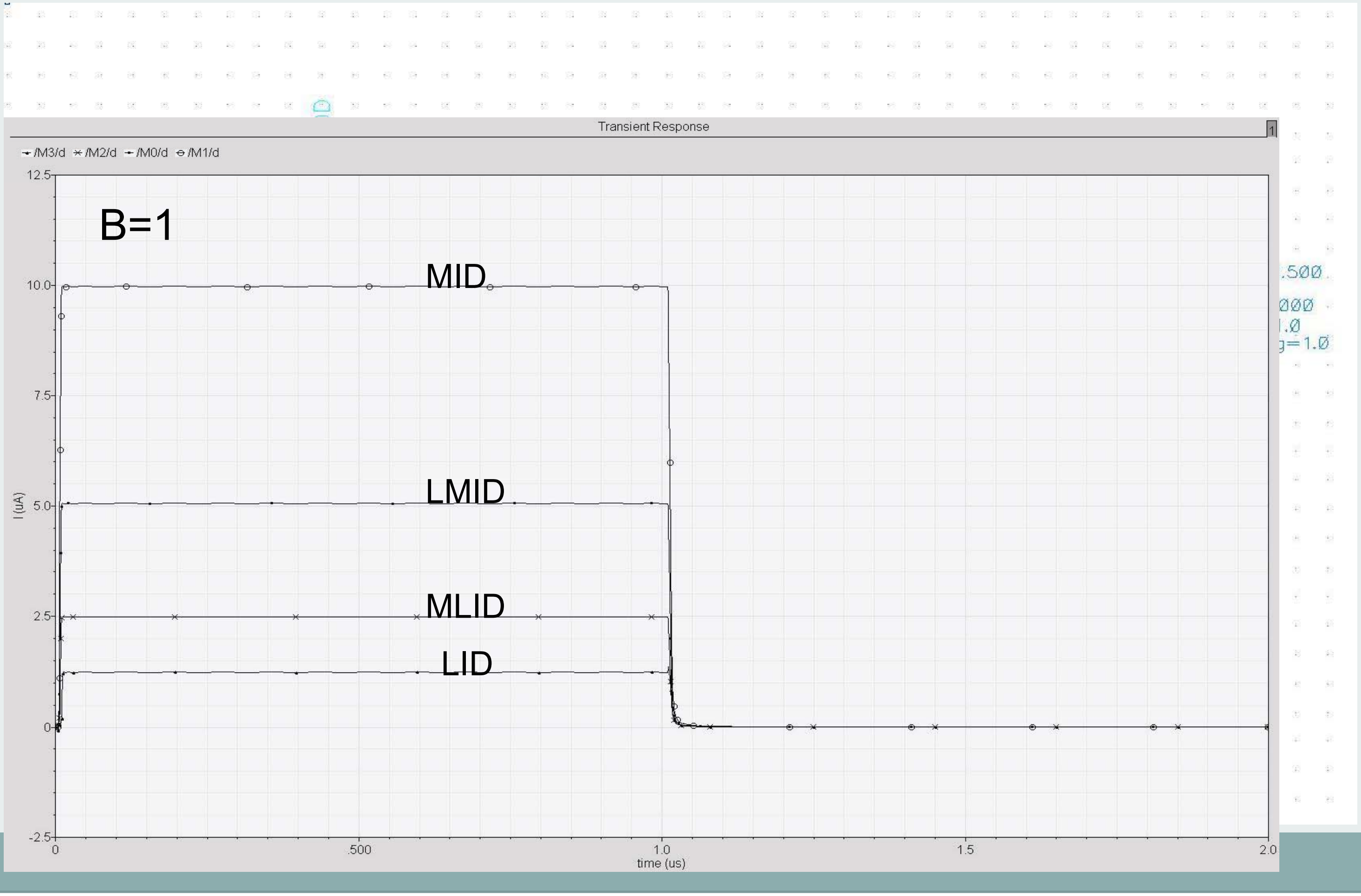
D-Latch

- ➢ Circuit
- ➢ Simulation Result



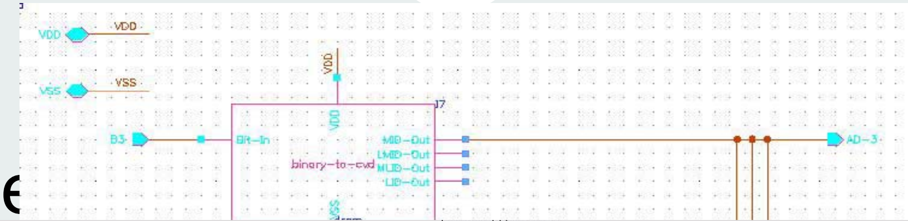
Binary To CVNS

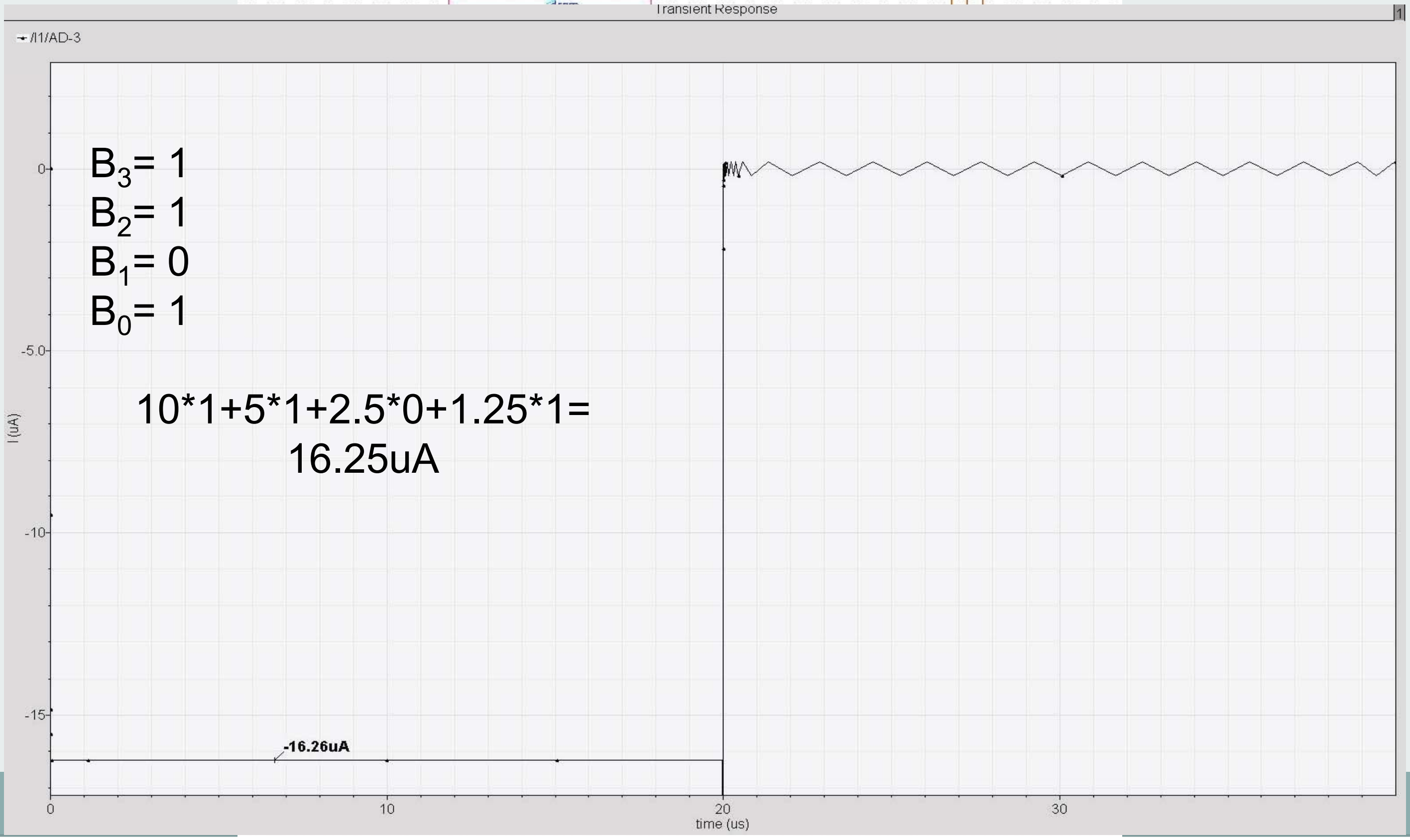
- Circuit
- Sim



Binary To CVNS (for 4-bit)

- Circuit
- Simulation Re



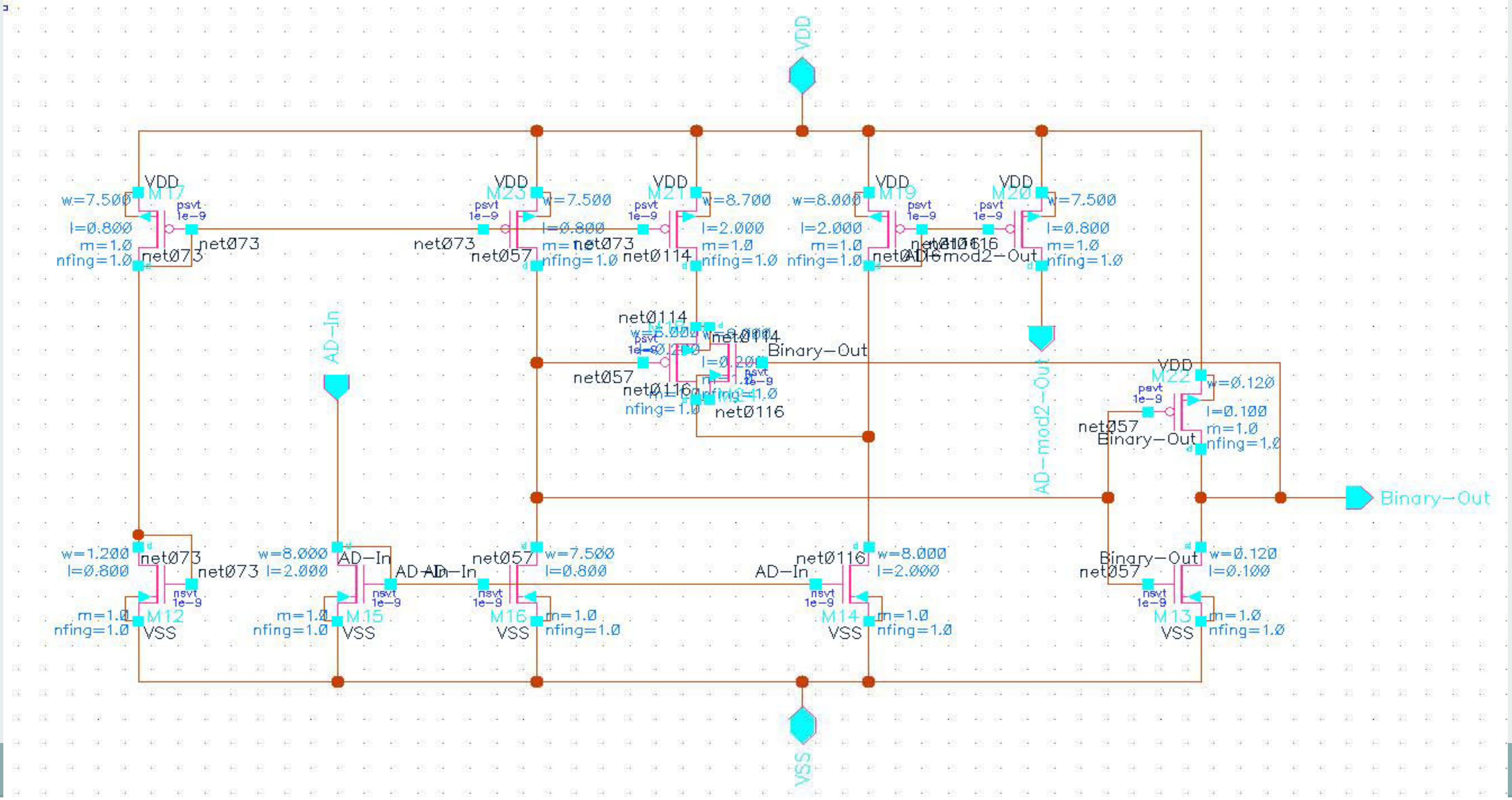


B_3= 1
B_2= 1
B_1= 0
B_0= 1

10*1+5*1+2.5*0+1.25*1=
16.25uA

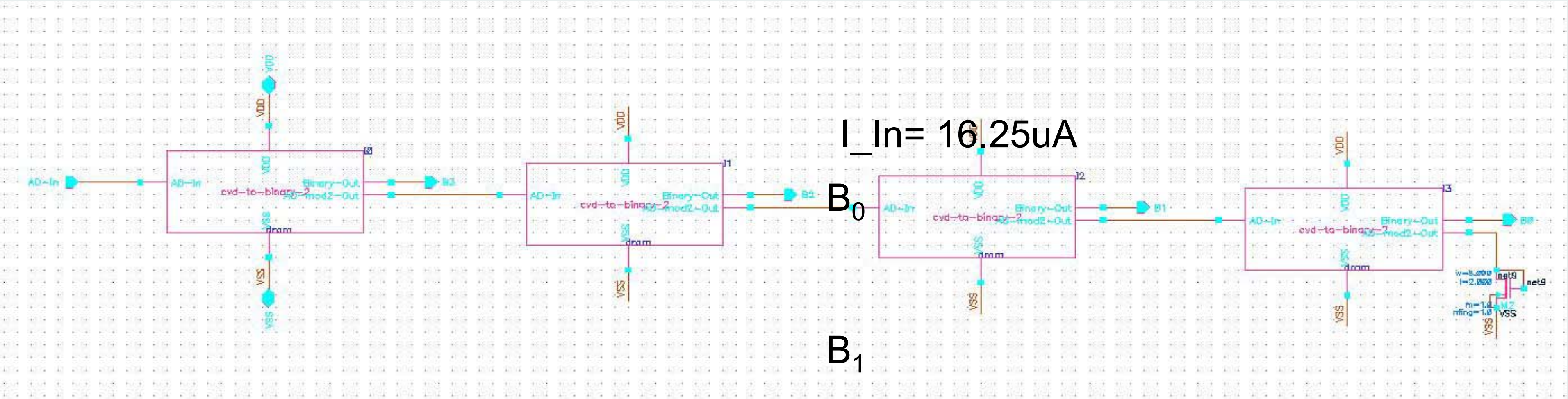
CVNS To Binary

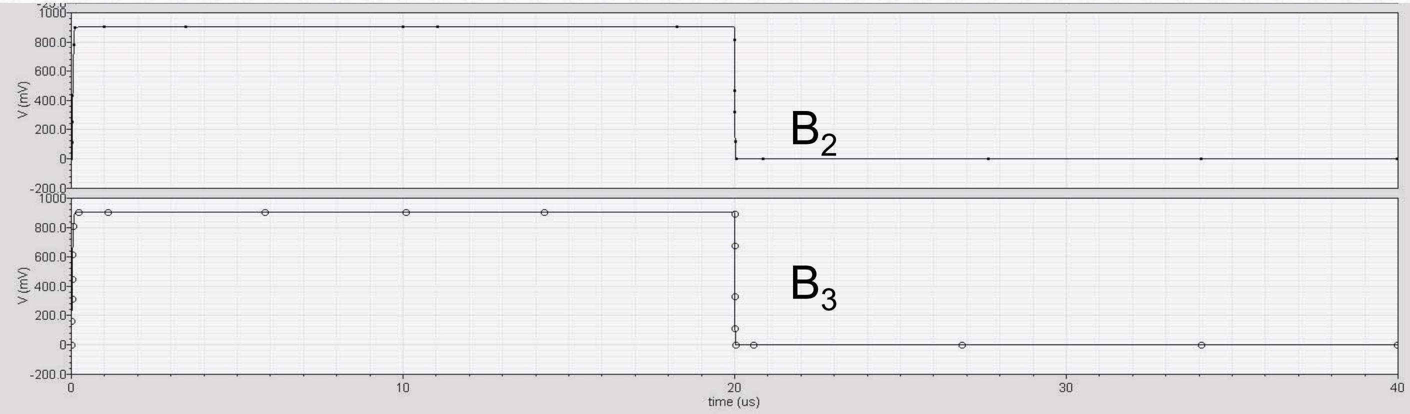
- Circuit



CVNS To Binary (for 4-bit)

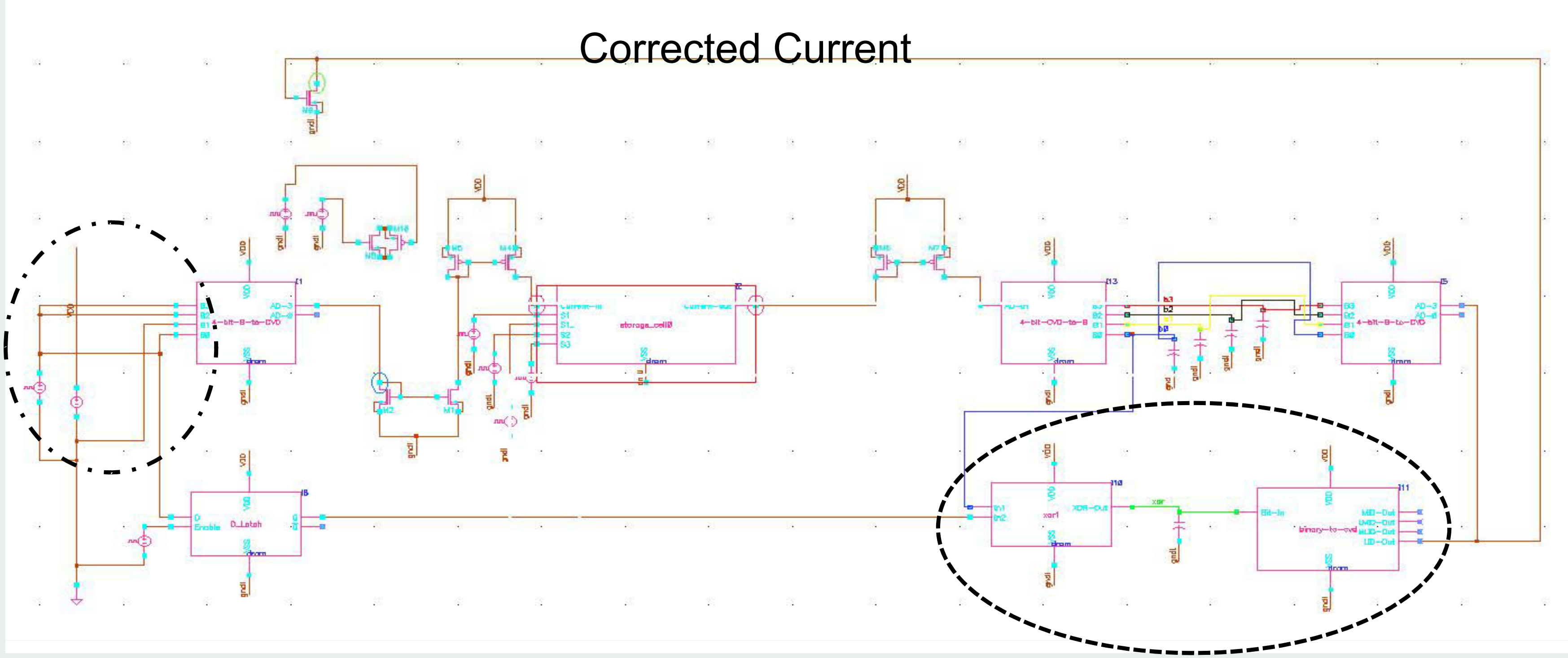
- Circuit





Refresh Circuit

- Simulated Circuit



Corrected Current

Refresh Circuit (Cont.)

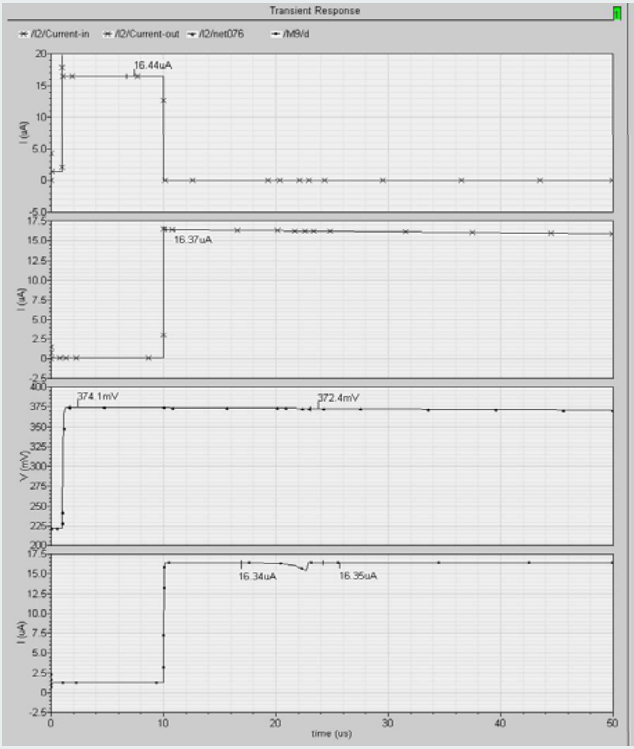
➢ Simulation Result

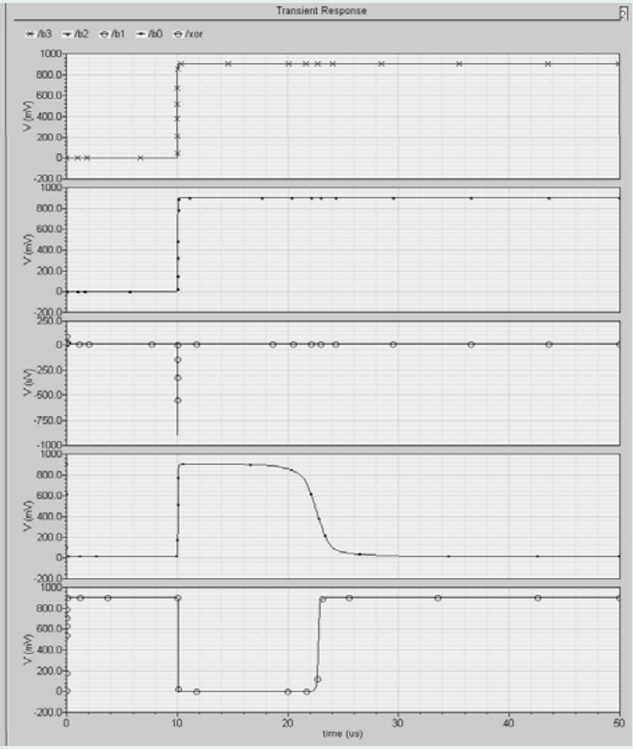
Current-In

Current-Out

V_C

Corrected-Current





B_3

B_2

B_1

B_0

XOR

Future Work

- Extend for 16-bit
- Draw the Layout